



Dokumentácia projektu

# Interpret jazyka IFJ14

10. decembra 2014

Autor: TODO, [xloginXX@stud.fit.vutbr.cz](mailto:xloginXX@stud.fit.vutbr.cz)  
Fakulta Informačních Technologí  
Vysoké Učení Technické v Brně

# Obsah

<b>1</b>	<b>Úvod</b>	<b>1</b>
<b>2</b>	<b>Riešenie projektu</b>	<b>1</b>
2.1	Interpret . . . . .	1
2.2	Algoritmy . . . . .	1
2.3	Práca v tíme . . . . .	1
<b>A</b>	<b>Metriky kódu</b>	<b>1</b>
<b>B</b>	<b>LL Gramatika</b>	<b>2</b>
<b>C</b>	<b>Precedenčná tabuľka</b>	<b>2</b>

# Notes

	Vymyslieť text . . . . .	1
	Sem dopíše Tomáš postupy postupy riešenia algoritmov etc. . . . .	1
	Aktualizovať metriky po dopísaní projektu . . . . .	1

# 1 Úvod

Vymyslieť text

## 2 Riešenie projektu

### 2.1 Interpret

### 2.2 Algoritmy

Sem dopíše Tomáš postupy postupy riešenia algoritmov etc.

### 2.3 Práca v tíme

## A Metriky kódu

Počet riadkov zdrojového textu: 12689 riadkov

Veľkosť statických dát: 6147B

Veľkosť spustiteľného programu: 13294B

Aktualizovať metriky po dopísaní projektu

## B LL Gramatika

PROGRAM	→ VAR_DECLR FUNC COMPOUND_STMT .
VAR_DECLR	→ var VAR_DEF
VAR_DECLR	→ $\epsilon$
VAR_DEF	→ id : type ; VAR_DEFN
VAR_DEFN	→ id : type ; VAR_DEFN
VAR_DEFN	→ $\epsilon$
FUNC	→ function id PARAM_DEF_LIST : type ; FORWARD FUNC
FUNC	→ $\epsilon$
FORWARD	→ forward ;
FORWARD	→ VAR_DECLR COMPOUND_SEMICOLON_STMT
PARAM_DEF_LIST	→ ( PARAMS_DEF )
PARAMS_DEF	→ id : type PARAMS_DEF_N
PARAMS_DEF	→ $\epsilon$
PARAMS_DEF_N	→ ; id : type PARAMS_DEF_N
PARAMS_DEF_N	→ $\epsilon$
TERM_LIST	→ ( TERMS )
TERMS	→ term TERMS_N
TERMS_N	→ , term TERMS_N
TERMS_N	→ $\epsilon$
COMPOUND_STMT	→ begin STMT_E end
COMPOUND_SEMICOLON_STMT	→ COMPOUND_STMT ;
STMT_LIST	→ $\epsilon$
STMT_LIST	→ ; STMT STMT_LIST
STMT_E	→ STMT STMT_LIST
STMT_E	→ $\epsilon$
STMT	→ id := EXPR
STMT	→ if EXPR then COMPOUND_STMT IF_N
STMT	→ while EXPR do COMPOUND_STMT
STMT	→ repeat STMT STMT_LIST until EXPR
STMT	→ COMPOUND_STMT
STMT	→ readln ( id )
STMT	→ write TERM_LIST
IF_N	→ else COMPOUND_STMT
IF_N	→ $\epsilon$

## C Precedenčná tabuľka

---

<sup>1</sup>Unárne mínus

	U- <sup>1</sup>	not	*	/	and	+	-	or	xor	<	>	<=	>=	=	<>	(	)	f	,	\$	var
U- <sup>1</sup>	H	R	R	R	R	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
not	R	H	R	R	R	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
*	S	S	R	R	R	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
/	S	S	R	R	R	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
and	S	S	R	R	R	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
+	S	S	S	S	S	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
-	S	S	S	S	S	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
or	S	S	S	S	S	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
xor	S	S	S	S	S	R	R	R	R	R	R	R	R	R	R	S	R	S	R	R	S
<	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
>	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
<=	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
>=	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
=	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
<>	S	S	S	S	S	S	S	S	S	R	R	R	R	R	R	S	R	S	R	R	S
(	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	H	S	H	E	S
)	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	E	R	E	R	R	E
f	E	E	E	E	E	E	E	E	E	E	E	E	E	E	E	H	E	E	E	E	E
,	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	H	S	H	E	S
\$	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	E	S	E	E	S
var	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	E	R	E	R	R	E